Flume-ng+hbase 整合

环境准备

Hadoop+hbase+zookeeper+flume-ng

配置介绍

以 master 机器作为 flume 数据的源、并将数据发送给 node1 机器上的 flume，最后 node1 机器上的 flume 将数据插入到 Hbase 中。

<一>Master 机器上的 example.conf 配置

在 master 的$FLUME_HOME/conf/目录下创建以下文件（文件名随便取），并做如下配置，这是数据的发送端：

```
agent.sources =baksrc
agent.channels=memoryChannel
agent.sinks =remotesink

agent.sources.baksrc.type = exec
agent.sources.baksrc.command = tail -F /home/test/data/data.txt
agent.sources.baksrc.checkperiodic = 1000
agent.sources.baksrc.channels=memoryChannel

agent.channels.memoryChannel.type = memory
agent.channels.memoryChannel.keep-alive = 30
agent.channels.memoryChannel.capacity = 10000
agent.channels.memoryChannel.transactionCapacity = 10000

agent.sinks.remotesink.type = avro
agent.sinks.remotesink.hostname =node1
agent.sinks.remotesink.port = 8888
agent.sinks.remotesink.channel= memoryChannel
```

<二>node1 机器上的 example.conf 配置

```
agent.sources = avrosrc
agent.channels = memoryChannel
agent.sinks = fileSink

agent.sources.avrosrc.type = avro
agent.sources.avrosrc.bind =node1
agent.sources.avrosrc.port =8888
agent.sources.avrosrc.channels = memoryChannel
agent.channels.memoryChannel.type = memory
agent.channels.memoryChannel.keep-alive = 30
agent.channels.memoryChannel.capacity = 10000
agent.channels.memoryChannel.transactionCapacity =10000

agent.sinks.fileSink.type =hbase
agent.sinks.fileSink.table =wpy
agent.sinks.fileSink.columnFamily =cf
agent.sinks.fileSink.column=charges
agent.sinks.fileSink.serializer =org.apache.flume.sink.hbase.RegexHbaseEventSerializer
agent.sinks.fileSink.channel = memoryChannel
```

启动 flume-ng

在 master 机器和 node1 机器上分别启动 flume 服务进程：

```
[root@master flume]$ bin/flume-ng agent
                                --conf conf
                                --conf-file conf/example.conf
                                --name agent
                                -Dflume.root.logger=INFO,console

[root@node1 flume]$ bin/flume-ng agent
                                --conf conf
                                --conf-file conf/example.conf
                                --name agent
                                -Dflume.root.logger=INFO,console
```

在 hbase 中创建好表

测试数据

vi test.sh

```
for i in {1..1000000}; do
    echo "test flume to Hbase $i" >> /home/test/data/data.txt;
    sleep 0.1;
done
```

Flume-ng+HBase 采集和存储日志数据

1.前提条件

Hadoop+HBase+Zookeeper+Flume-ng

2.解析日志程序

①AccessLog.java

```java
package com.tcloud.flume;
public class AccessLog {
    private String clientIp;
    private String clientIndentity;
    private String remoteUser;
    private String dateTime;
    private String request;
    private String httpStatusCode;
    private String bytesSent;
    private String referer;
    private String userAgent;
    public String getClientIp() {
        return clientIp;
    }
    public void setClientIp(String clientIp) {
        this.clientIp = clientIp;
    }
    public String getClientIndentity() {
        return clientIndentity;
    }
    public void setClientIndentity(String clientIndentity) {
        this.clientIndentity = clientIndentity;
    }
    public String getRemoteUser() {
        return remoteUser;
    }
    public void setRemoteUser(String remoteUser) {
        this.remoteUser = remoteUser;
    }
    public String getDateTime() {
        return dateTime;
    }
    public void setDateTime(String dateTime) {
        this.dateTime = dateTime;
    }
    public String getRequest() {
        return request;
    }
    public void setRequest(String request) {
        this.request = request;
    }
    public String getHttpStatusCode() {
        return httpStatusCode;
    }
    public void setHttpStatusCode(String httpStatusCode) {
        this.httpStatusCode = httpStatusCode;
    }
    public String getBytesSent() {
        return bytesSent;
    }
    public void setBytesSent(String bytesSent) {
        this.bytesSent = bytesSent;
    }
    public String getReferer() {
        return referer;
    }
    public void setReferer(String referer) {
        this.referer = referer;
    }
    public String getUserAgent() {
        return userAgent;
    }
    public void setUserAgent(String userAgent) {
        this.userAgent = userAgent;
    }
}
```

②AccessLogParser.java

```
package com.tcloud.flume;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class AccessLogParser {
    /**
     * 日志格式
     * 11.52.10.49 - - [17/Sep/2015:11:35:21 +0800] "GET /webapp
HTTP/1.1" 302 - "-" "Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.120
Safari/537.36"
     */
    private static String pattern = "^([\\d.]+) (\\S+) (\\S+)
\\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(.+?)\" (\\d{3}) (\\d+|-)
\"([^\"]+)\" \"([^\"]+)\"";
    private static Pattern p = Pattern.compile(pattern);
    public static AccessLog parse(String line){
        Matcher matcher = p.matcher(line);
          if (matcher.matches()){
              AccessLog accessLog = new AccessLog();
              accessLog.setClientIp(matcher.group(1));
```

```
                accessLog.setClientIndentity(matcher.group(2));
                accessLog.setRemoteUser(matcher.group(3));
                accessLog.setDateTime(matcher.group(4));
                accessLog.setRequest(matcher.group(5));
                accessLog.setHttpStatusCode(matcher.group(6));
                accessLog.setBytesSent(matcher.group(7));
                accessLog.setReferer(matcher.group(8));
                accessLog.setUserAgent(matcher.group(9));
                return accessLog;
            }
            return null;
        }
}
```

③AsyncHbaseLogEventSerializer.java

```
package com.tcloud.flume;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;
import org.apache.flume.Context;
import org.apache.flume.Event;
import org.apache.flume.conf.ComponentConfiguration;
import org.apache.flume.sink.hbase.HbaseEventSerializer;
import org.apache.hadoop.hbase.client.Increment;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.client.Row;
import org.apache.hadoop.hbase.util.Bytes;

public class AsyncHbaseLogEventSerializer  implements
HbaseEventSerializer {

    //列族
    private byte[] colFam="cf".getBytes();

    private Event currentEvent;
    @Override
    public void initialize(Event event, byte[] colFam) {

            this.currentEvent = event;
            this.colFam = colFam;
    }
    @Override
    public void configure(Context context) {}
    @Override
    public void configure(ComponentConfiguration conf) {
    }
    @Override
    public List<Row> getActions() {
            // Split the event body and get the values for the columns
            String eventStr = new String(currentEvent.getBody());
            AccessLog cols = AccessLogParser.parse(eventStr);
            String req = cols.getRequest();
            String reqPath = req.split(" ")[1];
            int pos = reqPath.indexOf("?");
         if (pos > 0) {
              reqPath = reqPath.substring(0,pos);
      }
         if(reqPath.length() > 1 && reqPath.trim().endsWith("/")){

           reqPath = reqPath.substring(0,reqPath.length()-1);
         }
      String req_ts_str = cols.getDateTime();
      Long currTime = System.currentTimeMillis();
      String currTimeStr = null;
      if (req_ts_str != null && !req_ts_str.equals("")){
                      SimpleDateFormat df = new
SimpleDateFormat("dd/MMM/yyyy:HH:mm:ss",Locale.US);
                         SimpleDateFormat df2 = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
                     try {

                             currTimeStr =
df2.format(df.parse(req_ts_str));

                             currTime =
df.parse(req_ts_str).getTime();

                      } catch (ParseException e) {

                             System.out.println("parse req time
error,using system.current time.");

                      }

            }
            long revTs = Long.MAX_VALUE - currTime;
            //行健根据自己需求设计
            byte[] currentRowKey = 
(UUIDGenerator.getUUID()+Long.toString(revTs)+ reqPath).getBytes();

            List<Row> puts = new ArrayList<Row>();
            Put putReq = new Put( currentRowKey);


             //增加列
             putReq.add( colFam,  "clientip".getBytes(), 
Bytes.toBytes(cols.getClientIp()));
             putReq.add( colFam,  "clientindentity".getBytes(), 
Bytes.toBytes(cols.getClientIndentity()));
             putReq.add( colFam,  "remoteuser".getBytes(), 
Bytes.toBytes(cols.getRemoteUser()));
             putReq.add( colFam,  "httpstatuscode".getBytes(), 
Bytes.toBytes(cols.getHttpStatusCode()));
             putReq.add( colFam,  "bytessent".getBytes(), 
Bytes.toBytes(cols.getBytesSent()));
             putReq.add( colFam,  "request".getBytes(), 
Bytes.toBytes(cols.getRequest()));
              putReq.add( colFam,  "referer".getBytes(), 
Bytes.toBytes(cols.getReferer()));
              putReq.add( colFam,  "datetime".getBytes(), 
Bytes.toBytes(currTimeStr));
              putReq.add( colFam,  "useragent".getBytes(), 
Bytes.toBytes(cols.getUserAgent()));

              puts.add(putReq);

              return puts;

     }
    @Override
     public List<Increment> getIncrements() {

                List<Increment> incs = new ArrayList<Increment>();

                return incs;

        }

    @Override
    public void close() {

         colFam = null;

         currentEvent = null;
    }

}
```

④UUIDGenerator.java

```
package com.tcloud.flume;

import java.util.UUID;

public class UUIDGenerator {

    public UUIDGenerator() {
    }
    /**
     * 获得一个 UUID
     * @return String UUID
     */
    public static String getUUID(){
        String s = UUID.randomUUID().toString();
        //去掉“-”符号
        return
s.substring(0,8)+s.substring(9,13)+s.substring(14,18)+s.substring(19
,23)+s.substring(24);
    }
    /**
     * 获得指定数目的 UUID
     * @param number int 需要获得的 UUID 数量
     * @return String[] UUID 数组
     */
    public static String[] getUUID(int number){
        if(number < 1){
            return null;
        }
        String[] ss = new String[number];
        for(int i=0;i<number;i++){
            ss[i] = getUUID();
        }
        return ss;
    } }
```

将上面的类导出成 jar 文件，放在 flume-ng 的 lib 目录下

3.通过 hbase 的 shell 建立 access_log 表，其中列族为 cf

4.配置 flume-ng

<一>数据源配置，监控日志产生，并发送给 agent

在 FLUME-NG 的安装目录的 conf 下建立 tomcatToHbase.conf

```
agent.sources =baksrc
agent.channels=memoryChannel
agent.sinks =remotesink

agent.sources.baksrc.type = exec
agent.sources.baksrc.command = tail -F /home/test/data/data.txt
agent.sources.baksrc.checkperiodic = 1000
agent.sources.baksrc.channels=memoryChannel

agent.channels.memoryChannel.type = memory
agent.channels.memoryChannel.keep-alive = 30
agent.channels.memoryChannel.capacity = 1000
agent.channels.memoryChannel.transactionCapacity = 1000

agent.sinks.remotesink.type = avro
agent.sinks.remotesink.hostname =spider-agent
agent.sinks.remotesink.port = 9999
agent.sinks.remotesink.channel= memoryChannel
```

<二>数据入库 hbase，接收收集的数据

在 FLUME-NG 的安装目录的 conf 下建立 tomcatToHbase.conf

```
agent.sources = avrosrc
agent.channels = memoryChannel
agent.sinks = fileSink

agent.sources.avrosrc.type = avro
agent.sources.avrosrc.bind =spider-agent
agent.sources.avrosrc.port =9999
agent.sources.avrosrc.channels = memoryChannel

agent.channels.memoryChannel.type = memory
agent.channels.memoryChannel.keep-alive = 30
agent.channels.memoryChannel.capacity = 1000
agent.channels.memoryChannel.transactionCapacity =1000

agent.sinks.fileSink.type = hbase
agent.sinks.fileSink.channel=memoryChannel
agent.sinks.fileSink.table = access_log
agent.sinks.fileSink.columnFamily =cf
agent.sinks.fileSink.batchSize=5
agent.sinks.fileSink.serializer
=com.tcloud.flume.AsyncHbaseLogEventSerializer
```

5.启动 flume-ng

在 master 机器和 node1 机器上分别启动 flume 服务进程：

```
[root@master flume]$ bin/flume-ng agent
                                --conf conf
                                --conf-file conf/tomcatToHbase.conf
                                --name agent
                                -Dflume.root.logger=INFO,console

[root@node1 flume]$ bin/flume-ng agent
                                --conf conf
                                --conf-file conf/tomcatToHbase.conf
                                --name agent
                                -Dflume.root.logger=INFO,console
```